1. A method comprising:

transferring data from a host memory to an Ethernet device; and

processing the data without sending the data from the host memory to an embedded memory associated with an adapter that includes the Ethernet device.

2. The method of claim 1 including forming protocol headers in the embedded memory.

3. The method of claim 1 including assigning descriptors to point to headers and data.

4. The method of claim 3 including assigning descriptors to point to headers and data in said embedded memory and host memory.

5. The method of claim 1 including computing checksums in firmware and in the Ethernet device.

6. The method of claim 1 including determining whether data in said host memory is larger than an Ethernet maximum transmit unit.

7. The method of claim 6 wherein if said data is larger than an Ethernet maximum transmit unit, placing said data in at least two different Ethernet packets.

8. The method of claim 7 including placing more headers in one Ethernet packet than in another Ethernet packet.

9. The method of claim 1 including forming a primary inbound window to receive data from a host and forming a secondary inbound window to receive data from said Ethernet device, said secondary inbound window having the same base address and limit as the primary inbound window.

10. The method of claim 1 including detecting the address of an access request from an Ethernet device and routing said request to the host memory or embedded memory based on the address.

11. An article comprising a medium storing instructions that enable a processor-based system to:

transfer data from a host memory to an Ethernet device; and

process the data without sending the data from the host memory to an embedded memory associated with an adapter that includes the Ethernet device.

12. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to form protocol headers in the embedded memory.

13. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to assign descriptors to point to headers and data.

14. The article of claim 13 comprising a medium storing instructions that enable a processor-based system to assign descriptors to point to headers and data in both said embedded memory and host memory.

15. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to compute checksums in firmware and in the Ethernet device.

16. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to determine whether data in said host memory is larger than an Ethernet maximum transmit unit.

17. The article of claim 16 comprising a medium storing instructions that enable a processor-based system

to, place said data in at least two different Ethernet packets if said data is larger than an Ethernet maximum transmit unit.

18. The article of claim 17 comprising a medium storing instructions that enable a processor-based system to place more headers in an Ethernet packet than in another Ethernet packet.

19. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to form a primary inbound window to receive data from a host and form a secondary inbound window to receive data from said Ethernet device, said secondary inbound window having the same base address and limit as the primary inbound window.

20. The article of claim 11 comprising a medium storing instructions that enable a processor-based system to detect the address of an access request from an Ethernet device and route said request to the host memory or embedded memory based on the address.

21. An adapter comprising:

a processor to communicate with a host system including a host memory;

an Ethernet device coupled to said processor; and

an embedded memory coupled to the processor to enable data to be transferred directly from the host memory to the Ethernet device without being copied to said embedded memory.

22. The adapter of claim 21 including descriptors that point to headers and data in said host memory and said embedded memory.

23. The adapter of claim 21 wherein said processor determines whether data in the host memory is larger than an Ethernet maximum transmit circuit.

24. The adapter of claim 23 wherein said processor places said data in at least two different Ethernet packets when said data is larger than said maximum transmit circuit.

25. The adapter of claim 24 wherein said processor places more headers in one Ethernet packet than in another Ethernet packet.

26. A system comprising:

a host processor;

a host memory coupled to said host processor;

a bridge coupled to said host processor; and

an adapter coupled to said bridge, said adapter including a processor, an embedded memory and an Ethernet device, said adapter to transfer data directly from said host memory to said Ethernet device without copying the data to the embedded memory.

27. The system of claim 26 wherein said bridge detects the address of an access request from said Ethernet device and routes said request to the host memory or embedded memory based on said address.

28. The system of claim 26 wherein said bridge includes a primary and secondary address translation unit, a primary memory coupled to the primary address translation unit and a secondary memory coupled to the secondary address translation unit.

29. The system of claim 28 wherein the primary inbound window of said primary memory is of the same size as the secondary inbound window of said secondary memory.

30. The system of claim 29 wherein said embedded memory is cacheable.